

In the Claims:

Please amend Claims 1 and 12; cancel Claims 23-26; and add new Claims 27-30, all as shown below. Applicant respectfully reserves the right to prosecute any originally presented or canceled claims in a continuing or future application.

1. (Currently Amended) A system for high availability clustering of a group of computer nodes, comprising:

a computer that allows ~~a user or an~~ application to access a set of resources of various resource types, including different application servers, within a cluster, wherein said resources and application servers are available at said computer or at another computer in the cluster, and wherein resources and application servers can be grouped by resource type within a pool of resources;

a cluster server that operates at said computer and that allows access to said set of resources;

a resource interface provided by said cluster server that provides an abstraction layer and allows the cluster server to receive uniform requests from the application and communicate with the requests to said set of resources;

a plurality of plugins that are plugged into the resource interface to provide a set of application-specific callbacks from the cluster server to the different resources ~~depending on the global events occurring within the group of computer nodes~~, wherein the system includes a plugin for each resource type, and wherein ~~each one of said plurality of plugins~~ plugin implements a resource API to encapsulate its particular resource type-specific behavior and to isolate the cluster server from that behavior while providing ~~provides access to a its pool of resources of a particular resource type, and wherein each pool of resources includes a plurality of resources of that particular type;~~

wherein additional plugins may be ~~included in~~ plugged into the resource interface for other resource types; and

wherein the system can be extended by adding additional computers with cluster servers and resource interfaces operating thereon.

2. (Original) The system of claim 1 wherein each of said cluster servers includes a heartbeat interface that provides heartbeat information to other cluster servers at said other application servers.
3. (Original) The system of claim 1 wherein the system is Java-based.
4. (Original) The system of claim 3 wherein the system includes a JNDI interface that provides an interface between the cluster server and a JNDI-compliant database.
5. (Original) The system of claim 1 wherein the system includes a cluster administration utility for accessing and administering the cluster server using remote method invocation calls.
6. (Original) The system of claim 1 wherein each resource has a resource type associated with it.
7. (Original) The system of claim 6 wherein resources are the object instances of their respective resource types.
8. (Original) The system of claim 1 wherein a resource is any of a computer, internet protocol address, disk, database, or file system or application.
9. (Original) The system of claim 1 wherein the cluster server defines resource groups that includes clusters of resources.
10. (Original) The system of claim 1 wherein the plugins include a WebLogic plugin.
11. (Original) The system of claim 1 wherein the plugins include a Tuxedo plugin.

12. (Currently Amended) A method for providing a high availability clustering framework system for a group of computer nodes, comprising the steps of:

allowing ~~a user or an~~ application to access, via a computer and a cluster server operating thereon, a set of resources of various resource types, including different application servers, within a cluster wherein said resources ~~being are~~ available at said computer or at another computer, and wherein resources and application servers can be grouped by resource type within a pool of resources;

providing a resource interface at said cluster server that provides an abstraction layer and allows the cluster server to receive uniform requests from the application and communicate with the requests to said set of resources via a plurality of plugins that are plugged into the resource interface;

~~to provide a set of application-specific callbacks from cluster server to resources depending on the global events occurring within the group of computer nodes, wherein each one of said plurality of plugins provides access to a pool of resources of a particular resource type, and wherein each pool of resources includes a plurality of resources of that particular type;~~

wherein the plurality of plugins are plugged into the resource interface to provide a set of application-specific callbacks from the cluster server to the different resources, wherein the system includes a plugin for each resource type, and wherein each plugin implements a resource API to encapsulate its particular resource type-specific behavior and to isolate the cluster server from that behavior while providing access to its pool of resources;

wherein additional plugins may be included in the resource interface for other resource types; and

wherein the system can be extended by adding additional computers with cluster servers and resource interfaces operating thereon.

13. (Original) The method of claim 12 wherein said cluster server includes a heartbeat interface provides heartbeat information to other cluster servers at said other application servers.

14. (Original) The method of claim 12 wherein the system is Java-based.
15. (Original) The method of claim 14 wherein the system includes a JNDI interface that provides an interface between the cluster server and a JNDI-compliant database.
16. (Original) The method of claim 12 wherein the system includes a cluster administration utility for accessing and administering the cluster server using remote method invocation calls.
17. (Original) The method of claim 12 wherein each resources has a resource type associated with it.
18. (Original) The method of claim 17 wherein resources are the object instances of their respective resource types.
19. (Original) The method of claim 12 wherein a resource is any of a computer, ip address, disk, database, or file system or application.
20. (Original) The method of claim 12 wherein the cluster server allows for clustering resources within a resource group.
21. (Original) The method of claim 12 wherein the plugins include a WebLogic plugin.
22. (Original) The method of claim 12 wherein the plugins include a Tuxedo plugin.
- 23-26. (Canceled).
27. (New) A system for high-availability clustering in a network, comprising:

a computer that receives requests from a client application to access one or a plurality of application servers of different types within a cluster, wherein the application servers are available either at the computer or at another computer within the cluster;

a cluster server that operates at the computer and provides access to the plurality of application servers, wherein the cluster server further comprises a resource interface that provides an abstraction layer and allows the cluster server to receive uniform requests from the client application and communicate the requests to the application servers; and

a plurality of plugins that can be plugged into the resource interface to provide application-specific callbacks from the cluster server to the different application servers, wherein the system includes a plugin for each application server type, and wherein each plugin implements a resource interface that encapsulates the particular resource type-specific behavior for that application server type, and isolates the cluster server from that behavior while still providing access to the application server.

28. (New) The system of claim 27 wherein, for each application server type, an appropriate plug-in is loaded at the time the first application server of a defined type is created, and wherein a handle is created to the specific resource instance, which can then be used by the cluster server in subsequent method calls.

29. (New) A method for high-availability clustering, comprising the steps of:

receiving requests at a computer from a client application to access one or a plurality of application servers of different types within a cluster, wherein the application servers are available either at the computer or at another computer within the cluster;

communicating the requests to a cluster server that operates at the computer and provides access to the plurality of application servers, wherein the cluster server further comprises a resource interface that provides an abstraction layer and allows the cluster server to receive uniform requests from the client application and communicate the requests to the application servers; and

using a plurality of plugins that can be plugged into the resource interface to provide application-specific callbacks from the cluster server to the different application servers, wherein the system includes a plugin for each application server type, and wherein each plugin implements a resource interface that encapsulates the particular resource type-specific behavior for that application server type, and isolates the cluster server from that behavior while still providing access to the application server.

30. (New) The method of claim 29 wherein, for each application server type, an appropriate plug-in is loaded at the time the first application server of a defined type is created, and wherein a handle is created to the specific resource instance, which can then be used by the cluster server in subsequent method calls.